
libpymath

Release 0.6.0

Oct 19, 2020

Contents

1	The Aim	3
1.1	Licencing	3
2	Installing the Package	5
3	Using the Package	7
4	Contents	9
4.1	libpymath	9
5	Indices and tables	15
	Python Module Index	17
	Index	19

Libpymath - a fast, general purpose Python math library

CHAPTER 1

The Aim

The aim of libpymath is to provide a simple and intuitive interface to complex mathematical routines, while maintaining a fast and efficient operation. The module will allow you to create powerful matrices, progress bars, neural networks and more with only a few lines of code.

1.1 Licencing

Libpymath is covered under an MIT license, so you can use all of the code as is for private, open source or commercial projects, as long as the copyright notice is retained at the top of any files used.

Installing the Package

To install the package, simply open your CLI of choice and run `pip install libpymath` to install the package. Hopefully there will be wheels provided for your operating system and Python version, however none exist the code must be built from source. This only takes a few seconds, however you will need a C or C++ compiler installed on your system.

If you would like to upgrade to the latest version of `libpymath`, you can run `pip install libpymath --upgrade` to upgrade an existing version installed on your system.

Another option for more experienced users is to clone the repository using the `git clone https://github.com/Pencilcaseman/LibPyMath.git` command in a directory of your choice. This will install the source code for you, allowing you to edit the code, add features, fix potential bugs (hopefully you won't find any) or simply see how it works.

CHAPTER 3

Using the Package

With libpymath installed, you can open a Python environment, such as IDLE, and start to write your first piece of code.

To import the module, add this to the top of your file:

```
import libpymath
```

Libpymath is commonly imported as `lpm`, which is shorter and easier to type. This change can be made easily:

```
import libpymath as lpm
```

The libpymath import allows you to access all of the features of libpymath, such as the matrix library or the progress bar library. For information on how to use these features, see below.

4.1 libpymath

4.1.1 libpymath package

Subpackages

libpymath.matrix package

Submodules

libpymath.matrix.matrix module

class libpymath.matrix.matrix.**Matrix**(*args, **kwargs)
Bases: object

T

See Matrix.transposed()

Returns Return the transpose of a matrix

cols

The number of columns of the matrix

Type return

copy()

Return an exact copy of a matrix

Returns Copied matrix

dot(other)

Compute the matrix-matrix product with another matrix

Parameters **other** – Matrix to compute matrix product with

Returns Result of matrix product calculation

dtype

The datatype of the matrix

Type return

fill (*fillType*, **args*)

Fill a matrix. Valid fills are:

SCALAR -> Fill with a single scalar value
ASCENDING -> Fill a matrix with ascending values, starting with 0
DESCENDING -> Fill a matrix with descending values, starting from (rows * cols) - 1
RANDOM -> Fill a matrix with random values between a given range (defaults to [-1, 1])

Parameters

- **fillType** – Method to use when filling the matrix
- **args** – Some fill methods accept parameters

Returns None

fillAscending ()

See Matrix.map(ASCENDING)

Returns None

fillDescending ()

See Matrix.map(DESCENDING)

Returns None

fillRandom (_min=None, _max=None)

See Matrix.map(RANDOM)

Returns None

fillScalar (x)

See Matrix.map(SCALAR)

Parameters **x** – Scalar value to fill with

Returns None

map (*mapType*)

Apply a function to every element of the matrix.

Valid functions are: SIGMOID TANH RELU LEAKY_RELU

D_SIGMOID D_TANH D_RELU D_LEAKY_RELU

Parameters **mapType** – Function to map with

Returns None

mapped (*mapType*)

See Matrix.map()

Parameters **mapType** – Function to map with

Returns Mapped matrix

mean ()

reshape (nr, nc)

Reshape a matrix by adjusting the rows and columns.

Parameters

- **nr** – New rows
- **nc** – New columns

Returns None

reshaped (*nr, nc*)

Reshape a matrix by adjusting the rows and columns and return the result.

Parameters

- **nr** – New rows
- **nc** – New columns

Returns Reshaped matrix

rows

The number of rows of the matrix

Type return

shape

The shape of the matrix in the form (rows, columns)

Type return

sum ()

threads

The number of threads the matrix is using

Type return

toList ()

Convert a Matrix into a 2d Python list

Returns 2d Python list

transpose ()

Transpose a matrix inplace.

Returns None

transposed ()

See Matrix.transpose()

Returns Return the transpose of a matrix

Module contents

Copyright 2020 Toby Davis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libpymath.progress package

Submodules

libpymath.progress.progress module

```
class libpymath.progress.progress.Progress (iterable=None, message=None, erase=False,  
                                           start=None, end=None, step=None, smooth-  
                                           ness=0.85)  
  
    Bases: object  
  
    static generateBar (fill, length, fillChar='', emptyChar='')  
  
    reset ()  
  
    update ()  
  
class libpymath.progress.progress.frange (start, end=None, step=None)  
    Bases: object
```

Module contents

libpymath.src package

Subpackages

libpymath.src.error package

Module contents

libpymath.src.matrix package

Module contents

Module contents

Module contents

Copyright 2020 Toby Davis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 5

Indices and tables

- search

I

- `libpymath`, [12](#)
- `libpymath.matrix`, [11](#)
- `libpymath.matrix.matrix`, [9](#)
- `libpymath.progress`, [12](#)
- `libpymath.progress.progress`, [12](#)
- `libpymath.src`, [12](#)
- `libpymath.src.error`, [12](#)
- `libpymath.src.matrix`, [12](#)

C

`cols` (*libpymath.matrix.matrix.Matrix* attribute), 9
`copy()` (*libpymath.matrix.matrix.Matrix* method), 9

D

`dot()` (*libpymath.matrix.matrix.Matrix* method), 9
`dtype` (*libpymath.matrix.matrix.Matrix* attribute), 10

F

`fill()` (*libpymath.matrix.matrix.Matrix* method), 10
`fillAscending()` (*libpymath.matrix.matrix.Matrix* method), 10
`fillDescending()` (*libpymath.matrix.matrix.Matrix* method), 10
`fillRandom()` (*libpymath.matrix.matrix.Matrix* method), 10
`fillScalar()` (*libpymath.matrix.matrix.Matrix* method), 10
`frange` (class in *libpymath.progress.progress*), 12

G

`generateBar()` (*libpymath.progress.progress.Progress* static method), 12

L

`libpymath` (module), 12
`libpymath.matrix` (module), 11
`libpymath.matrix.matrix` (module), 9
`libpymath.progress` (module), 12
`libpymath.progress.progress` (module), 12
`libpymath.src` (module), 12
`libpymath.src.error` (module), 12
`libpymath.src.matrix` (module), 12

M

`map()` (*libpymath.matrix.matrix.Matrix* method), 10
`mapped()` (*libpymath.matrix.matrix.Matrix* method), 10
`Matrix` (class in *libpymath.matrix.matrix*), 9

`mean()` (*libpymath.matrix.matrix.Matrix* method), 10

P

`Progress` (class in *libpymath.progress.progress*), 12

R

`reset()` (*libpymath.progress.progress.Progress* method), 12
`reshape()` (*libpymath.matrix.matrix.Matrix* method), 10
`reshaped()` (*libpymath.matrix.matrix.Matrix* method), 11
`rows` (*libpymath.matrix.matrix.Matrix* attribute), 11

S

`shape` (*libpymath.matrix.matrix.Matrix* attribute), 11
`sum()` (*libpymath.matrix.matrix.Matrix* method), 11

T

`T` (*libpymath.matrix.matrix.Matrix* attribute), 9
`threads` (*libpymath.matrix.matrix.Matrix* attribute), 11
`toList()` (*libpymath.matrix.matrix.Matrix* method), 11
`transpose()` (*libpymath.matrix.matrix.Matrix* method), 11
`transposed()` (*libpymath.matrix.matrix.Matrix* method), 11

U

`update()` (*libpymath.progress.progress.Progress* method), 12